PATENT ABSTRACTS OF JAPAN

(11)Publication number : 10-334056
(43)Date of publication of application : 18.12.1998

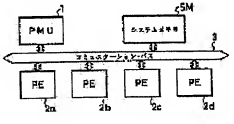
(51)Int.Cl. G06F 15/16

(21)Application number : 09-145384
(22)Date of filing : 03.06.1997

(71)Applicant : SONY CORP
(72)Inventor : TAMURA YUKIHIRO

(54) MULTIPROCESSOR SYSTEM

(57)Abstract:
PROBLEM TO BE SOLVED: To provide a multiprocessor system capable of describing a program on the basis of use of a processor in a form not to specialize the description to a processor with specified hardware structure and describing the program on the basis of the use of an optional number of the processors.
SOLUTION: The system is provided with a means to assign a process to a virtual processor, a means to assign processors 2a to 2d mounted on the system to the virtual processor, a means SM, 1 to store the virtual processor and the processors 2a to 2d assigned to the virtual processor by relating them and a means to enable the processors 2a to 2d stored in relation to the virtual processors to execute the virtual processors according to a fact that either of the virtual processors are called.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号
特開平10－334056
(43)公開日　平成10年(1998)12月18日

(51)Int.Cl.⁶	識別記号	ＦＩ	
Ｇ０６Ｆ　15/16	３７０	Ｇ０６Ｆ　15/16	３７０Ｎ

審査請求　未請求　請求項の数１　ＯＬ　（全５頁）

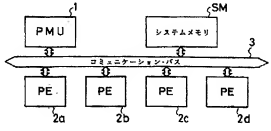
(21)出願番号　特願平9－145384

(22)出願日　平成９年(1997)６月３日

(71)出願人　000002185
ソニー株式会社
東京都品川区北品川６丁目７番35号
(72)発明者　田村　征大
東京都品川区北品川６丁目７番35号　ソニー株式会社内
(74)代理人　弁理士　松隈　秀盛

(54)【発明の名称】　マルチプロセッサ・システム

(57)【要約】
【課題】　アプリケーションソフトウェアのレベルで、プロセッサの使用を前提としたプログラムの記述を、特定のハードウェア構成のプロセッサに特化しない形で行うことができ、しかも任意の数のプロセッサの使用を前提として行うことができるようにしたマルチプロセッサ・システムを提供する。
【解決手段】　プロセスを仮想プロセッサに割り当てる手段と、仮想プロセッサに、システムに実装されたプロセッサ２を割り当てる手段と、仮想プロセッサとそれに割り当てられたプロセッサ２とを関連付けて記憶する手段ＳＭ、１と、いずれかの仮想プロセッサが呼び出されることに応じて、それに関連付けて記憶されたプロセッサ２に、当該仮想プロセッサを実行させる手段とを備えている。



【特許請求の範囲】
【請求項１】　システムにおいて実行すべきプロセスを仮想的なプロセッサに割り当てる手段と、
前記仮想的なプロセッサに、システムに実装されたプロセッサを割り当てる手段と、
前記仮想的なプロセッサと該プロセッサに割り当てられた前記実装されたプロセッサとを関連付けて記憶する手段と、
いずれかの前記仮想的なプロセッサが呼び出されることに応じて、該プロセッサに関連付けて記憶された前記実装されたプロセッサに、該仮想的なプロセッサを実行させる手段とを備えたことを特徴とするマルチプロセッサ・システム。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】本発明は、複数のプロセッサを結合し、統一したＯＳ（オペレーティングシステム）のもとでハードウェア資源及びソフトウェア資源を共用した並列処理システムであるマルチプロセッサ・システムに関し、特に、アプリケーションソフトウェアのレベルで、プロセッサの使用を前提としたプログラムの記述を、特定のハードウェア構成のシステムに特化しない形で行うことができ、しかも任意の数のプロセッサの使用を前提として行うことができるようにしたものに関する。
【０００２】
【従来の技術】従来、パーソナル・コンピュータ等で採用されているＳＭＰ（対称型マルチプロセッサ）システムをはじめとする各種マルチプロセッサ・システムにおいて、ハードウェア資源であるプロセッサは、全てＯＳの管理下に置かれていた。
【０００３】
【発明が解決しようとする課題】このようにプロセッサがＯＳの管理下に置かれていると、アプリケーションソフトウェアのレベルでは、プロセッサの使用を前提としたプログラムの記述を行うことは、特定のハードウェア構成のシステムに特化した形で行う以外には不可能である。
【０００４】また、このように特定のハードウェア構成のシステムに特化した形でプログラムを記述する場合には、システムのハードウェア構成に変更が生じる毎にプログラムを書き直さなければならないので、時間，労力，コストのロスが大きくなる。
【０００５】更に、このように特定のシステムに特化した形でプログラムを記述する場合には、システムに実装されているプロセッサ数よりも多い数のプロセッサの使用を前提としたプログラムを記述することはできないので、そのことがアプリケーションソフトウェアを作成する上での制約となっていた。
【０００６】本発明は上述の点に鑑みてなされたもので、アプリケーションソフトウェアのレベルで、プロセッサの使用を前提としたプログラムの記述を、特定のハードウェア構成のシステムに特化しない形で行うことができ、しかも任意の数のプロセッサの使用を前提として行うことができるようにしたマルチプロセッサ・システムを提供しようとするものである。
【０００７】
【課題を解決するための手段】本発明に係るマルチプロセッサ・システムは、システムにおいて実行すべきプロセスを仮想的なプロセッサ（以下、仮想プロセッサと呼ぶ）に割り当てる手段と、仮想プロセッサに、システムに実装されたプロセッサ（以下、物理プロセッサとも呼ぶ）を割り当てる手段と、仮想プロセッサとそれに割り当てられた物理プロセッサとを関連付けて記憶する手段と、いずれかの仮想プロセッサが呼び出されることに応じて、それに関連付けて記憶された物理プロセッサに、当該仮想プロセッサを実行させる手段とを備えたことを特徴としている。
【０００８】このマルチプロセッサ・システムでは、実装されたプロセッサとは別の「仮想プロセッサ」の概念を導入しており、システムにおいて実行すべきプロセスを、この仮想プロセッサに割り当てる。仮想プロセッサには物理プロセッサが割り当てられ、仮想プロセッサとそれに割り当てられた物理プロセッサとは関連付けて記憶される。そして、いずれかの仮想プロセッサが呼び出されると、それに関連付けて記憶された物理プロセッサに、当該仮想プロセッサを実行させる。
【０００９】このように、プロセッサへのアクセスを仮想プロセッサに対して行えば、システムに実装されたプロセッサにプロセスを実行させることができる。換言すれば、特定のハードウェア構成とは無関係な仮想的なプロセッサの使用を前提としたプログラムの記述を行えば、システムに実装されたプロセッサにアクセスすることができるようになる。
【００１０】従って、アプリケーションソフトウェアのレベルで、特定のハードウェア構成のプロセッサに特化しない形で記述したプログラムにより、プロセッサにアクセスすることができるようになる。
【００１１】また、システムに実装されたプロセッサ数よりも多い数の仮想プロセッサにプロセスを割り当てることにより、システムに実装されたプロセッサ数よりも多い数のプロセッサの使用を前提としたプログラムを記述することもできるようになる。
【００１２】
【発明の実施の形態】以下、添付図面を参照して本発明の実施例を詳細に説明する。
《１．システムの構成例》図１は、本発明に係るマルチプロセッサ・システムの構成の一例を概念的に示す。このシステムには、ＯＳやアプリケーションソフトウェア等を実行するための複数のプロセッサエレメント（P

E）2（図では一例として4基のPE2a～PE2d）の他に、プロセッサ・マネージメント・ユニット（PMU）1が設けられている。PMU1は、仮想プロセッサを管理するための一種のコプロセッサである。

【0013】各PE2とPMU1とは、コミュニケーション・バス3を介して接続されている。コミュニケーション・バス3は、接続されている任意のエレメント間で多ビットの情報のやり取りが可能な双方向バスである。システムの主記憶装置であるシステムメモリSMも、このコミュニケーション・バス3を介して各PE2に接続されることにより、各PE2に共有されている。

【0014】〈1a．PMUとPETLB〉このマルチプロセッサ・システムでは、仮想プロセッサと、物理プロセッサである各PE2とに、それぞれ固有のID（識別ラベル）が与えられている。仮想プロセッサのIDを「VEID」，物理プロセッサPE2のIDを「PEID」と呼ぶ。

【0015】VEIDは、後出の〈3b．仮想プロセッサの生成〉で述べるように、システムの起動後新たに仮想プロセッサを生成するときに，OSによってその仮想プロセッサに与えられたものである。PEIDは、システムの構成に応じてハードウェアによって与えられたものであり，変更することはできない。

【0016】PMU1（図1）内には、仮想プロセッサのVEIDとその仮想プロセッサに割り当てられた（この割り当てについても後出の〈3b．仮想プロセッサの生成〉で述べる）物理プロセッサPE2のPEIDとを関連付けて記憶させることのできる記憶領域（テーブル）が用意されている。このテーブルを、プロセッサエレメントTLB（PETLB）と呼ぶ。

【0017】図2は、このPETLBの一例を概念的に示す。PETLBは同図Aのように複数のエントリを持っており、各エントリ（但し図の最上部のシステムエントリ8を除く）は同図Bのようにいくつかのフィールドに分かれている。

【0018】これらのフィールドのうち、VaIidフィールド4は，そのエントリに有効なデータが存在することを示すためのフィールドであり、Lockフィールド5は、そのエントリの内容を変更することが禁止されていることを示すためのフィールドである。同一エントリのVEIDフィールド6とPEIDフィールド7とには、仮想プロセッサのVEIDとその仮想プロセッサに割り当てられた物理プロセッサPE2のPEIDとがそれぞれ格納される。

【0019】各エントリに記憶される内容は、OSによってシステムメモリSM（図1）内で管理されている「プロセッサ・エレメント管理テーブル」（仮想プロセッサと物理プロセッサとの対応を定義するいわば仮想プロセッサの名簿）の一部分のコピーである。PETLBは一種の連想記憶装置になっており、VEIDを与えたとき、そのVEIDの仮想プロセッサに割り当てられた物理プロセッサPE2のPEIDを引き出せるようになっている。

【0020】尚、システムエントリ8は、後述のシステム・プロセッサに対する物理プロセッサPE2の割り当て情報を管理するための特別なエントリである。

【0021】PETLBのこうした構造は、一般的なマイクロプロセッサの仮想記憶方式に置いて用いられるテーブルと似ている。

【0022】〈1b，システム・プロセッサ〉「システム・プロセッサ」は、仮想プロセッサの中の一つであり、システム全体に関わるエラーやイベントを処理するための特別な仮想プロセッサである。システム外部からの割り込みや、仮想プロセッサの管理に関するエラー等は、このシステム・プロセッサで発生したエラーとして処理される。

【0023】システム・プロセッサ以外の一般の仮想プロセッサについてのエントリの初期化は，通常はソフトウェアによってシステムメモリSM内のプロセッサ・エレメント管理テーブルで行わなければならないが，システム・プロセッサだけは、後出の〈3a．システムの初期化〉で述べるように、システムのリセット時にハードウェアによってPETLBでシステムエントリ8が初期化される。

【0024】また、PETLBのシステムエントリ8以外のエントリは、後出の《2．仮想プロセッサへのアクセス》で述べるようにプロセッサ・エレメント管理テーブル中の他のエントリと入れ替えられることがあるが、システムエントリ8だけは、この入れ替えからは保護された位置にある。換言すれば，システム・プロセッサだけには常に物理プロセッサPE2が割り当てられており、システムエントリ8からその物理プロセッサPE2を検索することができる。但し、システム・プロセッサ以外の物理プロセッサPE2を割り当てるは，システムエントリ8自体を書き換えることについていても変更することが可能である。

【0025】《2．仮想プロセッサへのアクセス》アプリケーションソフトウェアがシステム中のプロセッサにアクセスする場合，（例えば現在プロセスを処理中のプロセッサとは別のプロセッサにプロセスを投入したり、現在プロセスを処理中のプロセッサとは別のプロセッサと通信を行ったりするような場合等）には，VEIDを用いて仮想プロセッサに対してアクセスするようにする。換言すれば，アプリケーションソフトウェアでは，VEIDを用いて，仮想プロセッサの使用を前提としたプログラムの記述を行うようにする。

【0026】現在プロセスを処理中のプロセッサ（即ち現在実行中の仮想プロセッサに割り当てられた物理プロセッサPE2（図1））の中で、このような別の仮想プロセッサに対するアクセスが発生すると，それが仮想プ

ロセッサの呼び出しイベントとしてコミュニケーション・バス3（図1）上に通知される。このとき、アクセスされた仮想プロセッサのＶＥＩＤも同時にバス3上に乗せられる。

【００２７】各物理プロセッサＰＥ２は、それぞれ実行中の仮想プロセッサのＶＥＩＤを記憶するための「ＶＥＩＤレジスタ」を持っており、このレジスタを参照することにより、現在どの仮想プロセッサに割り当てられているかを判別することができる。物理プロセッサＰＥ２は、バス3上に乗せられたＶＥＩＤと現在割り当てられている仮想プロセッサのＶＥＩＤとが一致した場合（即ち自分が呼び出された仮想プロセッサである場合）、アクセスに対して直接応答する。

【００２８】一方、ＰＭＵ１（図1）は、ＰＥＴＬＢ（図2）の各エントリから、バス3を介して送られたＶＥＩＤを格納しているＶＥＩＤフィールド６を検索し、そのＶＥＩＤフィールドと同じエントリ中のＰＥＩＤフィールド７から、当該仮想プロセッサに割り当てられた物理プロセッサＰＥ２のＰＥＩＤを得る。そして、いずれの物理プロセッサＰＥ２もアクセスに対して直接応答しなかったときには、得られたＰＥＩＤを持つ物理プロセッサＰＥ２に対して、仮想プロセッサの入れ替えのための割り込み要求を発生する。

【００２９】呼び出された仮想プロセッサに割り当てられた物理プロセッサＰＥ２では、この割り込みにより、ＶＥＩＤレジスタ内のＶＥＩＤの書き換えと、例外ハンドラ（例外処理を記述したプログラム）によるプロセスの入れ替えとが行われる。

【００３０】尚、ＰＭＵ１がＰＥＴＬＢの各エントリを検索した際に、バス3を介して送られたＶＥＩＤを格納しているＶＥＩＤフィールド６がみつからなかった場合には、前述のシステム・プロセッサに例外が発生する。この場合には、ＯＳが例外ハンドラによってＰＥＴＬＢのエントリの入れ替えを行い、その後改めて仮想プロセッサの呼び出しが行われる。

【００３１】《3．ソフトウェアとの連携による仮想プロセッサ管理の例》

（3ａ．システムの初期化）システムがリセットされると、各物理プロセッサＰＥ２は初期化されて待機状態になる。ＰＥＴＬＢ（図2）は、システムエントリ８が初期化され、それ以外のエントリが全て無効化される。

【００３２】すべての初期化処理が終了すると、ＰＭＵ１（図1）がシステム・プロセッサを呼び出す。この呼び出しは、《2．仮想プロセッサへのアクセス》で述べた物理プロセッサＰＥ２に対する割り込み要求によって行われる。

【００３３】起動したシステム・プロセッサでは、ＯＳのカーネル（核）の実行が開始され、システムメモリＳＭ（図1）内のプロセッサ・エレメント管理テーブルの初期化が行われる。

【００３４】（3ｂ．仮想プロセッサの生成）その後、ＯＳのその他の部分やアプリケーションソフトウェアが起動されて、それぞれのプロセスが新しい仮想プロセッサに割り当てられる。この処理は、プロセスが、ＯＳの用意したプログラムである「仮想プロセッサ生成タスク」をシステム・コール等で呼び出すことによって行われる。

【００３５】呼び出された仮想プロセッサ生成タスクは、プロセスを割り当てられた仮想プロセッサにＶＥＩＤを与え、その仮想プロセッサに物理プロセッサＰＥ２（図1）を割り当てる。そして、そのＶＥＩＤとＰＥＩＤと当該プロセスの実行に必要な情報とを、システムメモリＳＭ（図1）内のプロセッサ・エレメント管理テーブルに記憶させる。このようにして、仮想プロセッサが生成される。

【００３６】尚、新しく生成する仮想プロセッサにどの物理プロセッサＰＥ２を割り当てるかは、ＯＳが各物理プロセッサＰＥ２の使用状況等をもとにして決定する。ＶＥＩＤで表現可能な範囲内で、システムに実装されている物理プロセッサＰＥ２の数よりも多い数の仮想プロセッサを生成することも可能であり、その場合には複数の仮想プロセッサに同一の物理プロセッサＰＥ２を共有させるようにする。

【００３７】（3ｃ．仮想プロセッサの実行）新しく生成された仮想プロセッサを実際に物理プロセッサＰＥ２上で実行させる処理は、仮想プロセッサの生成処理とは独立かつ並行的に行われる。

【００３８】そのために、システムには、タイマ割り込みのような外部割り込みが定期的にかかるようになっている。この仕組みは、単一のプロセッサに時分割によりマルチタスクを行わせるためのものと同じである。

【００３９】タイマ割り込みはシステム全体に関係するイベントなので、それが発生すると、ＰＭＵ１が物理プロセッサＰＥ２に対する割り込み要求によって自動的にシステム・プロセッサを呼び出す。この割り込み処理を記述したプログラムである割り込みハンドラから、ＯＳの用意したプログラムである「仮想プロセッサ管理タスク」が呼び出されるようになっている。

【００４０】呼び出された仮想プロセッサ管理タスクは、システムメモリＳＭ内のプロセッサ・エレメント管理テーブルから、次のタイマ割り込みまでの一定期間物理プロセッサＰＥ２を割り当てて実行する仮想プロセッサを、各プロセス間の優先順位等をもとに決定する。こうして決定された仮想プロセッサが、《2．仮想プロセッサへのアクセス》で述べたようアプリケーションソフトウェアによってアクセスされることにより、その仮想プロセッサの実行が開始される。

【００４１】以上のように、このマルチプロセッサ・システムによれば、アプリケーションソフトウェアのレベルで、物理プロセッサＰＥ２のハードウェア構成に特化

しない形で記述したプログラムにより，物理プロセッサＰＥ２にアクセスすることができるようになる。

【００４２】また、システムに実装されている物理プロセッサＰＥ２の数よりも多い数の仮想プロセッサにプロセスを割り当てることにより、ＶＥＩＤで表現可能な範囲内で、物理プロセッサＰＥ２の数よりも多い数のプロセッサの使用を前提としたプログラムを記述することもできるようになる。

【００４３】尚、以上の実施例では、主記憶装置（システムメモリＳＭ）を単一のバス（コミュニケーション・バス３）を介して各プロセッサ（ＰＥ２）に共有させる単一バス方式の共有メモリ形（密結合形）マルチプロセッサに本発明を適用しているが，その他の方式（例えば多重バス方式，階層バス方式，マルチポートメモリ方式等）の共有メモリ形マルチプロセッサや，あるいはメッセージ交換形（疎結合形）マルチプロセッサに本発明を適用するようにしてもよい。

【００４４】また、本発明は、以上の実施例に限らず，本発明の要旨を逸脱することなく、その他様々の構成をとりうることはもちろんである。

【００４５】

【発明の効果】以上のように、本発明に係るマルチプロセッサ・システムによれば、アプリケーションソフトウェアのレベルで、特定のハードウェア構成のプロセッサに特化しない形で記述したプログラムにより，プロセッサにアクセスすることができる。これにより，システムにおけるプロセッサのハードウェア構成に変更が生じた場合にも、アプリケーションソフトウェアのプログラムを書き直す必要がなくなるので、時間，労力，コストの節約につながる。

【００４６】また、システムに実装されたプロセッサ数よりも多い数のプロセッサの使用を前提としたプログラムを記述することができるので、アプリケーションソフトウェアを作成する上での制約を減らすことができる。

【図面の簡単な説明】

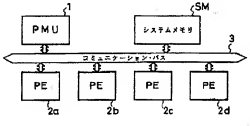
【図１】本発明に係るマルチプロセッサ・システムの構成の一例を示す図である。

【図２】プロセッサエレメントテーブルの一例を示す図である。

【符号の説明】

１…プロセッサ・マネージメント・ユニット、　２ａ，２ｂ，２ｃ，２ｄ…プロセッサエレメント、　３…コミュニケーション・バス、　ＳＭ…システムメモリ、　４…Ｖａｌｉｄフィールド、　５…Ｌｏｃｋフィールド、　６…ＶＥＩＤフィールド、　７…ＰＥＩＤフィールド、　８…システムエントリ

【図１】



【図２】